

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

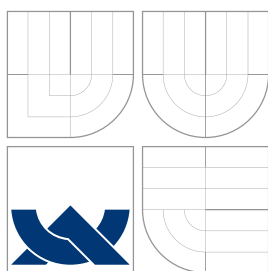
MODELOVÁNÍ EMOCÍ V KOMUNIKAČNÍM AGENTU

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

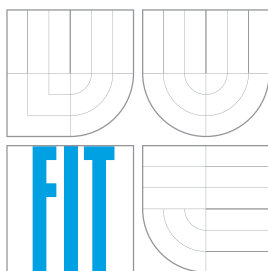
AUTOR PRÁCE  
AUTHOR

Bc. MARTIN SIVÁK

BRNO 2007



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **MODELOVÁNÍ EMOCÍ V KOMUNIKAČNÍM AGENTU**

MODELLING EMOTIONS IN COMMUNICATION AGENTS

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. MARTIN SIVÁK**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**doc. RNDr. PAVEL SMRŽ, Ph.D.**

BRNO 2007

# Modelování emocí v komunikačním agentu

---

## Modelling Emotions in Communication Agents

**Vedoucí:** Smrž Pavel, doc. RNDr., Ph.D., UPGM FIT VUT

**Oponent:** Černocký Jan, doc. Dr. Ing., UPGM FIT VUT

**Přihlášen:** Sivák Martin, Bc.

### Zadání:

1. Seznamte se s metodami a postupy používanými v současných komunikačních agentech.
2. Nastudujte literaturu věnovanou modelování emocí.
3. Implementujte systém, který bude schopen komunikovat s člověkem na libovolné téma a modelovat vybrané chování.
4. Vyhodnoťte výsledky práce na základě vhodně zvolených metrik.

### Část požadovaná pro obhajobu SP:

1. Prototyp systému s omezenou funkcionalitou, který bude základem pro další vývoj.

**Kategorie:** Umělá inteligence

### Literatura:

- Manning, C. D., Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999, ISBN 0-262-13360-1
-

## Licenční smlouva

---

**Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.**

Souhlasím s uveřejněním práce **ihned** po odevzdání.

---

## Abstrakt

Tato práce se zabývá problémy existujících komunikačních agentů a možnostmi jejich vylepšení. Hlavní důraz je kladen na schopnosti porozumění přirozenému jazyku a modelování emocí během konverzace. Ve druhé části je také představena implementace založená na prezentovaných poznatcích, včetně experimentálního vyhodnocení úspěšnosti.

## Klíčová slova

komunikační agent, zpracování jazyka, modelování emocí, konverzace, umělá inteligence

## Abstract

This work deals with current chatterbot systems. It describes problems and possibilities of improvement with emphasis on natural language processing and emotion modeling during conversation. There is an implementation, based on the described knowledge, introduced in the second part of the thesis, also with experimental success rate evaluation.

## Keywords

chatterbot, natural language processing, emotion modeling, conversation, artificial intelligence

## Citace

Martin Sivák: Modelování emocí v komunikačním agentu, diplomová práce, Brno, FIT VUT v Brně, 2007

# Modelování emocí v komunikačním agentu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Smrže a uvedl jsem všechny zdroje, ze kterých jsem čerpal.

.....

Martin Sivák  
17. května 2007

## Poděkování

Tímto bych chtěl poděkovat dobrovolníkům, kteří obětovali svůj čas pro vytvoření testovacích dat jmenovitě Petru Kolářovi, Tomáši Janouškovi, Alexi Butinovi a Marcelu Mašláňové.

© Martin Sivák, 2007.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Turingův test . . . . .	3
1.2	Využití výsledků a cíle práce . . . . .	4
<b>2</b>	<b>Existující řešení komunikačních agentů</b>	<b>5</b>
2.1	Eliza . . . . .	5
2.2	Alice a A.I.M.L. . . . .	6
2.3	MegaHAL . . . . .	6
2.4	Loebnerova cena . . . . .	7
<b>3</b>	<b>Možnosti vylepšení komunikačních agentů</b>	<b>8</b>
3.1	Znalosti okolního světa . . . . .	8
3.2	Lepší porozumění komunikaci . . . . .	8
3.3	Emoce . . . . .	8
<b>4</b>	<b>Modelování emocí</b>	<b>9</b>
4.1	OCC . . . . .	9
4.2	Neural Emotion Eliciting System . . . . .	10
4.3	Cathexis . . . . .	11
<b>5</b>	<b>Modelování osobnosti</b>	<b>13</b>
5.1	OCEAN . . . . .	13
5.2	Jungova teorie a MBTI . . . . .	14
<b>6</b>	<b>Zpracování jazyka</b>	<b>15</b>
6.1	Zjednoznačnění významu slov . . . . .	15
6.1.1	Metody založené na znalostech . . . . .	15
6.1.2	Metody s učitelem . . . . .	16
6.1.3	Metody bez učitele . . . . .	16
6.1.4	Další metody pro zpřesnění . . . . .	17

6.2	Identifikace větné stavby . . . . .	17
6.3	Určení tématu konverzace . . . . .	18
6.3.1	Metody využívající externí znalosti . . . . .	19
6.3.2	Metody využívající vektory příznaků . . . . .	19
<b>7</b>	<b>Implementace vlastního agenta</b>	<b>20</b>
7.1	Požadavky . . . . .	20
7.2	Popis navržené architektury . . . . .	20
7.3	Podpůrné rutiny . . . . .	20
7.3.1	DbTree . . . . .	22
7.3.2	Ukládání výsledků . . . . .	22
7.3.3	Lexikální analýza . . . . .	22
7.3.4	Gramatická pravidla . . . . .	23
7.3.5	CKY parser . . . . .	24
7.4	Příprava dat . . . . .	24
7.5	Zpracování větné struktury . . . . .	25
7.5.1	Určení slovních druhů . . . . .	25
7.5.2	Určení hranic mezi větami . . . . .	25
7.5.3	Rozpoznání významu slov . . . . .	26
7.5.4	Větná stavba . . . . .	26
7.6	Detekce a zpracování emocí . . . . .	29
7.6.1	Emoční model . . . . .	29
7.6.2	Třídy emocí . . . . .	29
7.6.3	Popis používané gramatiky . . . . .	30
7.6.4	Zpracování na úrovni slov . . . . .	30
7.6.5	Zpracování na úrovni celé věty . . . . .	31
7.6.6	Sémantické zpracování . . . . .	32
7.7	Model osobnosti . . . . .	32
7.8	Sestavení odpovědi . . . . .	33
7.8.1	Algoritmus hledání odpovědi . . . . .	34
7.9	Zobrazení odpovědi a vizualizace emocí . . . . .	34
7.10	Rozšíření pro účely testování . . . . .	35
7.10.1	Vzdálený přístup . . . . .	35
7.10.2	Vizualizace větné stavby . . . . .	36
<b>8</b>	<b>Testy a vyhodnocení</b>	<b>37</b>
8.1	Uživatelské testy . . . . .	37
<b>9</b>	<b>Závěr</b>	<b>40</b>



# Kapitola 1

## Úvod

Již několik desetiletí jsou zkoumány způsoby, jak vytvořit systém, který bude schopen komunikovat s člověkem přirozenějším způsobem. Aby bylo možné tohoto cíle dosáhnout, je potřeba prozkoumat mnoho oblastí jako zpracování signálů, rozpoznávání mluvené řeči, zpracování přirozeného jazyka, umělá inteligence, psychologie, multimédia nebo získávání a uchovávání znalostí.

Ve výzkumu těchto oblastí se angažovalo již velké množství známých osobností jako například Alan Turing, Joseph Weizenbaum, Kenneth Colby, Paul Malish a mnoho dalších. Přesto je výzkum ještě stále téměř na začátku a v této práci se nejdříve pokusím představit výsledky, kterých již bylo dosaženo a možnosti jejich využití při implementaci komunikačního agenta. Zde naváží na výsledky získané již v rámci semestrálního projektu. V dalších částech popíši jedno z možných řešení pro implementaci komunikačního agenta schopného emotivně reagovat na uživatelský vstup.

### 1.1 Turingův test

Už v roce 1950 popsal profesor Alan Turing [16] jeden ze způsobů, jakým rozhodnout, zda je stroj schopen inteligentní komunikace.

Princip Turingova testu je založen na imitační hře, kde porota komunikuje se dvěma skrytými subjekty a snaží se rozlišit, který je který. Subjekty se přitom mohou snažit porotu zmást. Pro potřeby umělé inteligence byla tato hra upravena tak, že subjekty jsou člověk a stroj a porota se na základě odpovědí na jí položené otázky snaží rozpoznat, který ze subjektů je strojem.

Stálým zdrojem diskuse mezi odborníky jsou některé nevýhody tohoto testu:

- I když může dojít k tomu, že stroj nebude během komunikace rozpoznatelný od člověka, stále to nemusí znamenat jeho inteligenci.
- Již dnes existují poměrně pokročilé techniky imitace komunikačních schopností člověka, tzv. „shallow AI“, které nelze označit za inteligenci.
- Také někteří lidé by mohli mít problémy projít tímto testem.
- A nakonec, inteligence nutně nevyžaduje rozvinuté komunikační schopnosti.

I když momentálně neexistuje žádný stroj, který by tímto testem úspěšně prošel, existují předpovědi, podle kterých by se tak mohlo stát kolem roku 2020[12].

## 1.2 Využití výsledků a cíle práce

Výsledky zde prezentované mohou mít význam pro množství praktických aplikací. Většina jich spadá do oblasti získávání informací, jako například systémy pro automatizované zjišťování preferencí zákazníků, různé typy průzkumů, vyhodnocení přepisů rozhovorů. Ve spojení s rozpoznáním mluvené řeči se potom množina praktických využití dále zvětšuje.

Cílem této práce je tedy nejprve zmapovat stav na poli komunikačních agentů a metod v nich využívaných. Dalším z cílů je navrhnout vlastní implementaci komunikačního agenta, založeného na poznatcích získaných v teoretické části.

Tento agent by měl být schopen identifikovat emoce ve vstupu, vyhodnotit jejich vztah k sobě a vygenerovat odpověď s použitím některého z popsáných algoritmů.

Důležitým úkolem je nakonec zhodnotit dopad použitých metod na míru úspěšnosti rozpoznání emocí a emoční reakce.

## Kapitola 2

# Existující řešení komunikačních agentů

Existující komunikační agenti (chatterbots [13]) využívají metody pro zpracování přirozeného jazyka, porovnávání vzorů a různé formy konečných automatů, aby docílili iluze inteligentní komunikace. Žádný z nich zatím nedisponuje skutečnou inteligencí a schopností porozumět tématu komunikace.

Níže popíšeme několik algoritmů u nejznámějších „Chatterbotů“ současnosti a poukáží na některé jejich slabiny.

### 2.1 Eliza

Jeden z prvních konverzačních programů vůbec byl navržen v roce 1966 Josephem Weizenbaumem, aby parodoval psychoterapeuta. Eliza kladla otázky založené na předchozí odpovědi uživatele s cílem rozvinutí informace dodané v předchozí komunikaci.

Algoritmicky byla Eliza velmi jednoduchý program, byla založena na parseru, který doplňoval slova do předpřipravených frází. I přes tuto jednoduchost byla v některých konverzacích skutečně přesvědčivá.

```
young woman: He says I'm depressed much of the time.
eliza: I am sorry to hear you are depressed.
young woman: It's true. I'm unhappy.
eliza: Do you think coming here will help you not to be unhappy?
young woman: I need some help, that much seems certain.
eliza: What would it mean to you if you got some help?
young woman: Perhaps I could learn to get along with my mother.
eliza: Tell me more about your family.
young woman: My mother takes care of me.
eliza: Who else in your family takes care of you?
```

Mezi nevýhody tohoto přístupu patří samozřejmě vysoká míra opakování a žádné skutečné znalosti o komunikačním partnerovi, okolí nebo samotné diskusi.

## 2.2 Alice a A.I.M.L.

Jedním z nejoblíbenějších chatterbotů současnosti je Alice (Artificial Linguistic Internet Computer Entity). Vznikla v roce 1995 a vyvíjí ji Dr. Richard S. Wallace a Alicebot komunita.

Alice je bot využívající techniku porovnávání vzorů. Znalostní databáze je tvořena vzorovými otázkami pro určité téma a odpověďmi na ně nebo akcí k vyvolání. Celá je uložena jako XML dokument v dialektu A.I.M.L.

Vzorová otázka je běžná věta s možností zástupných znaků. Akcemi, které je možné vyvolat, jsou textový výstup, spuštění jiné akce, změna kontextu, podmíněné vyvolání jiné akce, případně zavolání nějakého programu. Dalšími důležitými akcemi jsou uložení a načtení proměnné.

**Ukázka A.I.M.L.**

```
<category>
<pattern>WHO IS HE</pattern>
<template><srai>WHOISHE <get name='he' /></srai></template>
</category>
```

```
<category>
<pattern>WHOISHE *</pattern>
<template>He is <get name='he' />.</template>
</category>
```

```
<category>
<pattern>WHOISHE UNKNOWN</pattern>
<template>I don't know who he is.</template>
</category>
```

Nevýhody tohoto přístupu jsou zřejmé. Agent postavený nad tímto jazykem také není schopen porozumět konverzaci a není schopen komunikovat o ničem jiném, než do něj jeho autor při „programování“ vložil.

Nicméně pro specializované aplikace typu informace o produktu, navigace na webové stránce, nebo jako jednoduchý expertní systém, jsou jeho schopnosti dostačující a běžně se využívá.

## 2.3 MegaHAL

Algoritmus tohoto agenta vznikl primárně jako příspěvek pro Loebnerovu cenu<sup>2.4</sup> a jako pokus představit chatterbota, který nebude používat klasický algoritmus porovnávání vzorů.

MegaHAL využívá dva Markovovy modely čtvrtého řádu a jednoduchou techniku identifikace klíčových slov ve vstupní větě. Během učení jsou oba modely naučeny pravděpodobnosti sekvencí slov (první směrem dopředu, druhý směrem zpět) a dojde ke spočítání četnosti výskytu jednotlivých slov pro využití v algoritmu pro klíčová slova.

Algoritmus pro hledání odpovědi potom postupuje tak, že odfiltruje často se vyskytující slova, zbývající převede („my“ na „yours“ a podobně) a použije jako vstup pro oba automaty. Každý z těchto automatů vlastně vygeneruje jednu polovinu věty. Těchto kandidátních odpovědí je vygenerováno několik stovek a poté se vybere odpověď obsahující nejvyšší množství informace.

Množství informace v odpovědi je vyjádřeno jako  $I(w|s) = -\log_2 P(w|s)$ , kde  $P(w|s)$  je pravděpodobnost výskytu slova  $w$  po sekvenci  $s$ .

Odpověď s nejvyšší mírou informace obvykle bývá tou nejpřekvapivější, jakou algoritmus umí „vymyslet“. Tento postup bohužel také způsobuje, že je MegaHAL velmi „dobrý“ v generování nesmyslných vět.

MegaHAL se je také schopný učit ze vstupu od uživatelů a v tom se skrývá jedna z jeho největších slabin. Citlivost na chyby v trénovacích datech.

### Ukázka komunikace s MegaHAL algoritmem

MegaHAL: WHEN WE BEGIN TO RESOLVE SOME OF MY THUMBS,  
SOMETHING WICKED THIS WAY COMES

User: what is the meaning of your last assertion ?

MegaHAL: HAVE YOU LEARNED SINCE WE LAST SPOKE?

User: I've learned I'm not experienced enough to modelize your brain system

MegaHAL: YOU'VE FINALLY CONFUSED ME.

User: I've learned nothing

MegaHAL: HAVE YOU LEARNED ANYTHING ABOUT INSANITY?

User: are you stupid

MegaHAL: I AM NOT STUPID! REALLY!

## 2.4 Loebnerova cena

V roce 1990 Dr. Hugh Loebner a Cambridge Center for Behavioral Studies uspořádali „Turingův test pro chatterboty“. Vítězem se měl stát agent, jehož komunikační schopnosti budou nejpodobnější lidem. Od té doby se tato událost koná každý rok.

Existují 3 ceny, které je možné získat. Výše zmíněnou cenu za největší podobnost k lidské komunikaci, cenu za nerozlišitelnost člověka od stroje při textovém rozhovoru a cenu za nerozlišitelnost člověka od stroje v komunikaci obsahující i vizuální a zvukové prvky. Poslední dvě ceny budou uděleny pouze jednou, ale zatím nikdo této úrovni nedosáhl.

## Kapitola 3

# Možnosti vylepšení komunikačních agentů

Všechny výše uvedené metody měly několik společných vlastností. Žádná nebyla schopna skutečného porozumění konverzaci a všechny ukázky komunikace postrádaly jakékoliv trvalější emoce (existovala pouze odpověď na aktuální vstup od uživatele). Tyto vlastnosti moc neprospívají úrovni konverzace, a proto stojíme před problémy jak zajistit alespoň základní porozumění textu, dosáhnout přirozenějšího projevu a zajistit konzistenci „názorů“ modelované entity.

### 3.1 Znalosti okolního světa

Problémem většiny dnešních metod je absence tzv. „common sense knowledge“. Jedná se o základní znalosti, nad kterými člověk obvykle nepřemýšlí, ale v počítačovém světě se musí mnohdy složitě modelovat.

Jedním z možných řešení je zapojit metody známé z programování obecných agentů jako například BDI logika a schopnost jednoduchého odvozování v rámci této techniky a definovaných znalostí o okolí (více například v knize *Inteligentní agenty*[10]).

### 3.2 Lepší porozumění komunikaci

Jedním z těžších úkolů, před kterými dnešní výzkum stojí, je porozumění psanému textu. Techniky používané pro tento problém jsou většinou založené na zpracování přirozeného jazyka (NLP) a to hlavně v oblasti zjednoznačnění významu slov ve vstupu, identifikace kontextu a zpracování sémantiky.

### 3.3 Emoce

Další z možností, jak vytvořit více realistickou entitu, je zapojit některý z emočních modelů. Výsledkem by měly být lepší reakce na podněty, přirozenější konverzace, a tudíž mnohem snazší akceptovatelnost technologie lidmi.

## Kapitola 4

# Modelování emocí

Emoce tvoří důležitý prvek v lidském rozhodování a chování. Proto je pravděpodobné, že stroje schopné emocím porozumět a následně je i projevovat by byly mnohem lépe přijaty lidmi, kteří s nimi komunikují.

Obecně můžeme emoční modely rozdělit na architektury, které pouze doplňují schopnosti agenta a systémy, kde emoce jsou přímo součástí rozhodovacího procesu. Ať už se jedná o kteroukoliv z těchto možností, úkolem emočního modelu je obvykle upozornit agenta na vstupy, které jsou v danou chvíli důležité pro agentovy cíle a měly by být zpracovány přednostně [11].

Obecný postup při modelování emocí má 5 kroků [3].

- klasifikace – vyhodnocení události, akce nebo objektu pro určení, které emoční kategorie budou ovlivněny;
- kvantifikace – výpočet intenzity pro ovlivnění kategorií z předchozího bodu;
- interakce – zkombinování aktuálního emocionálního stavu s ovlivněním vypočítaným v předchozích bodech;
- mapování – převedení složitého emočního stavu na jednodušší reprezentaci nebo méně skupin (má smysl například u OCC modelu, kde je 22 emocí);
- vyjádření/zobrazení – grafické zobrazení, případně vliv na rozhodování a chování modelované entity;

Existuje několik různých modelů pro sledování emočního stavu entity. Jeden z nich, často používaný jako základ pro odvozování, je teoretický model OCC [2]. Dalšími modely jsou například Nemesys [7] nebo Cathexis [17].

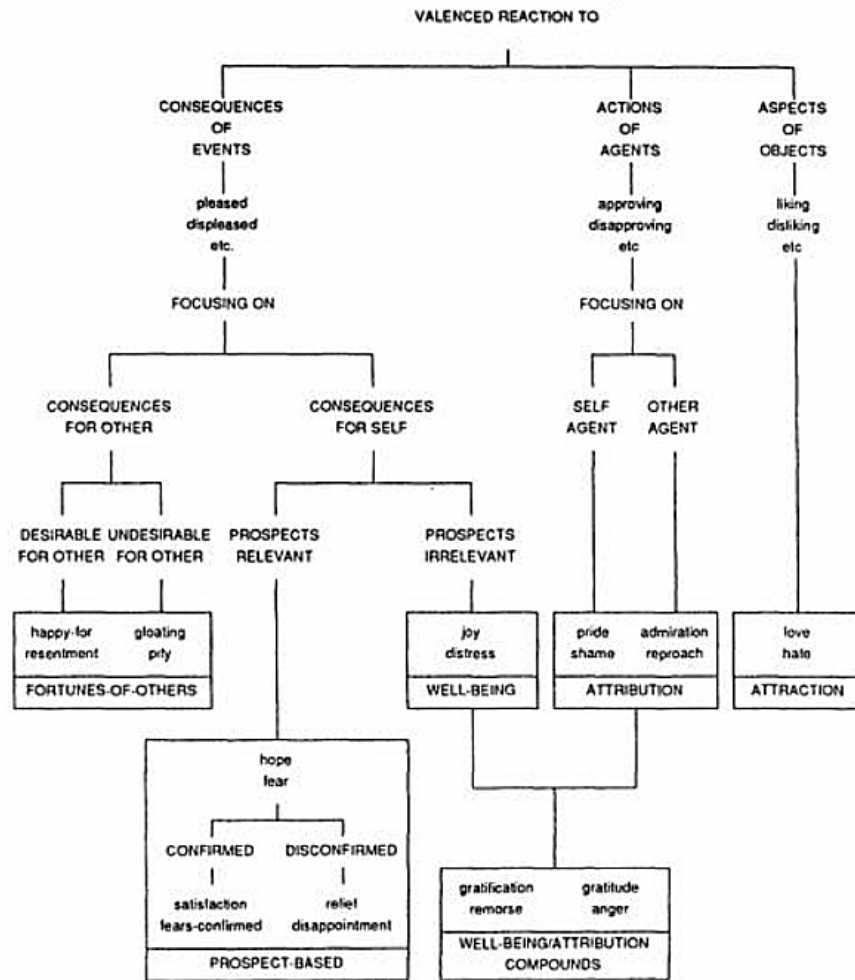
### 4.1 OCC

Model OCC [2], nazvaný podle svých tvůrců (Ortony, Clore, Collins), je považován za „standard“ při modelování emocí. Pro některé úlohy je ale příliš rozsáhlý a pro některé se nehodí, například díky tomu, že nespecifikuje vazby na rozhodovací proces agenta.

OCC rozděluje emoce do tří skupin:

- emoce ovlivněné pozorovanou událostí
- emoce ovlivněné akcí okolních agentů
- emoce ovlivněné vlastnostmi nějakého objektu

Tyto tři kategorie se dále dělí dokud není rozlišeno, která z 22 emocí bude aktivována.



Obrázek 4.1: OCC - struktura emocí (převzato z [2])

Následně je určena intenzita a provede se interakce s ostatními emocemi stejné kategorie. Posledním krokem je aplikace emočního stavu na výrazové prostředky agenta.

## 4.2 Neural Emotion Eliciting System

Jeden z relativně nových modelů, také známý jako Nemesys [7], využívá neuronovou síť pro modelování šesti základních emocí ( joy/anger/fear/sadness/disguist/surprise).



Jako vstup pro modelování slouží několik skupin vstupů:

- interní stav – týká se věcí jako hlad, únava;
- osobnost – Nemesys využívá model OCEAN[1] pro modelování osobnosti;
- stav agenta – v této jednotce se zpracovávají problémy vztahů, znalostí, důvěry, cíle a morálních principů;
- události okolí;
- statické a dynamické plány agenta.

Vstupy ze skupin stavu agenta a události okolí jsou nejdříve zpracovány hodnotícím modulem (appraisal unit) a poté spolu s údaji z prvních třech skupin předány neuronové síti, která má výpočte výsledný emoční stav.

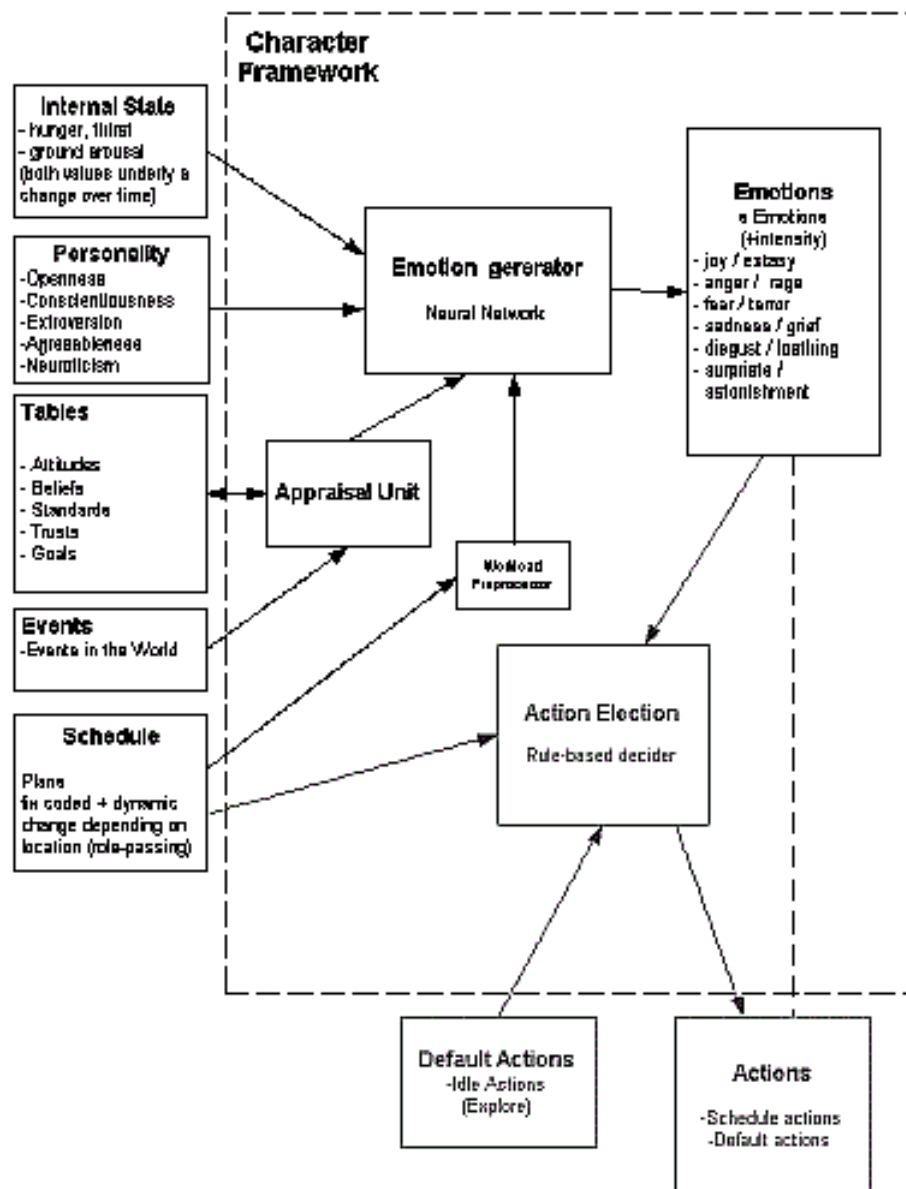
Získaný stav je možné dále využít, například v rozhodovacím nebo vizualizačním modulu agenta.

### 4.3 Cathexis

Model Cathexis [17] vytvořený Juanem Velazquezem a inspirovaný Minského dílem The Society Of Mind obsahuje dvě části: generátor emocí a řízení chování.

Generátor emocí je svou konstrukcí podobný neuronové síti. Každý z výstupů zpracovává vnější a vnitřní podněty (včetně ostatních výstupů) a v případě, že vnitřní hodnota překročí daný práh, aktivuje emoci, za kterou je zodpovědný. Vnitřní potenciál je shora omezený a v případě, že nedochází k žádnému relevantnímu podnětu, tak postupně klesá. Tato architektura umožňuje i generování nálad pomocí výstupů s nízkým prahem a nízkou excitační hranicí.

Část zodpovědná za řízení chování rozhoduje o dalším postupu agenta na základě vnitřních a vnějších podnětů, aktuálního stavu emocí a agentových plánů. Možnostem dalšího postupu jsou přiřazeny váhy podle již zmíněných vlastností a chování s nejvyšší celkovou váhou bude provedeno.



Obrázek 4.2: Nemesys - struktura (převzato z [7])

## Kapitola 5

# Modelování osobnosti

K emočnímu modelu, který vyjadřuje aktuální stav modelované entity, je vhodné přidat ještě model osobnosti, který bude popisovat vlastnosti nepatřící mezi emoce, jako například schopnost se učit, vnímat okolí nebo rychlost reakcí.

Existuje mnoho modelů používaných psychology ke klasifikaci osobnosti.

### 5.1 OCEAN

V poslední době se prosazuje hlavně model OCEAN, také známý jako Big5, a jeho různé deriváty. Popisuje vlastnosti, u kterých je empiricky zjištěno, že jsou stále i s narůstajícím věkem a mohou být částečně dědičné.

Název tohoto modelu je odvozen od jednotlivých skupin vlastností používaných pro popis osobnosti:

- Openness - Do této skupiny patří vlastnosti jako fantazie, umělecké citění, pocity, činy, hodnoty a ideály.
- Conscientiousness - Možný název je také sebekontrola a sdružuje kompetentnost, spořádanost, odhodlání, disciplinovanost a podobné.
- Extraversion - Tato složka modelu se zaměřuje na asertivitu, pozitivní emoce, aktivitu nebo třeba vyhledávání dobrodružství.
- Agreeableness - Také známá jako nezávislost a obsahuje důvěru, přímost, altruismus, přizpůsobivost nebo skromnost.
- Neuroticism - Skupina zodpovědná za zlost, deprese, impulzivitu a zranitelnost jedince.

Tyto skupiny jsou zastoupeny v každé osobnosti různou měrou, nejedná se tudíž o typy, ale spíše o dimenze.

## 5.2 Jungova teorie a MBTI

Jiným přístupem zabývajícím se osobností, může být teorie Karla Junga. Byla publikována již v roce 1923 a využívá se v testu na MBTI. Tvrdí, že osobnost lze popsat dvěma přístupy (introverze, extroverze) a čtyřmi funkcemi (přemýšlení, cítění, vnímání a intuice). Podle této teorie každý má všech šest popsaných rysů, ale jeden přístup a jedna funkce jsou dominantní.

Na základě této teorie vznikl model MBTI, který klasifikuje osobnost do 16 možných typů. Podle [14] lze tyto typy dále rozdělit do čtyř kategorií podle možností implementace (simulace nebo napodobení) dominantní a pomocné funkce.

## Kapitola 6

# Zpracování jazyka

Jak bylo vidět na příkladech u existujících chatterbotů, jedním z velkých problémů je porozumění uživatelskému vstupu. Protože toto téma je velmi rozsáhlé, pokusím se zde alespoň nastínit způsoby, jakými je možné vylepšit existující chatterboty.

### 6.1 Zjednoznačnění významu slov

Podle mého názoru jednou z nejdůležitějších oblastí pro vedení smysluplné konverzace je samotné porozumění významu jednotlivých slov. Jedná se o složitý problém, jelikož lidská řeč obsahuje velké množství nejednoznačností na téměř všech úrovních zpracování <sup>1</sup>.

Problém zjednoznačnění významu se dá zařadit mezi klasifikační problémy a lze využít známé techniky strojového učení pro jeho řešení. I když prozatím neexistuje žádné se 100% účinností, bylo navrženo několik metod, které se maximální úspěšnosti alespoň snaží přiblížit.

#### 6.1.1 Metody založené na znalostech

Metody tohoto typu využívají externí zdroje znalostí jako jsou různé slovníky, encyklopedie nebo sémantické sítě. O zjednoznačnění významu slov se poté pokoušejí, za pomoci znalostí získaných z těchto zdrojů, několika způsoby:

- nalezení podobností mezi zpracovávaným kontextem a definicí ve externích zdrojích;
- pozice v sémantické síti (například anglické „table“ má blíže ke kategorii „chair“ ve smyslu nábytek, než k „chair“ ve smyslu předseda).
- známý význam slova v již známém slovním obratu nebo skupině slov.
- a další

Do této skupiny metod patří několik algoritmů, o kterých je vhodné se zmínit:

---

<sup>1</sup>nejednoznačnosti na úrovních lexikálního zpracování, syntaxe i sémantiky slov a vět

## Leskův algoritmus

Počítá velikost překryvu slovníkových definicí pro všechny možné významy slov ve větě. Protože při větším množství slov ve větě počet kombinací extrémně narůstá, využívá se při výpočtu simulované žíhání.

Existuje také zjednodušená varianta tohoto algoritmu, která počítá překryv významu slova a aktuálního kontextu. Tento přístup je rychlejší a bývá i úspěšnější.

## Algoritmy s využitím encyklopedií

Tyto algoritmy jsou založeny na znalostech o vazbách mezi sémantickými kategoriemi. Je nutné mít bázi znalostí o tzv. common sense knowledge (zdravý selský rozum).

Celý postup by se poté dal popsat jako několik kroků:

- získat možné významy otevřených slov;
- nalézt sémantické vazby mezi možnými významy;
- vybrat ty významy, které odpovídají vybraným vazbám.

### 6.1.2 Metody s učitelem

V případě, že máme k dispozici data, kde jsou otevřené kategorie slov předem označovány svými významy v použitém kontextu, můžeme využít některý z algoritmů pro strojové učení s učitelem.

Obecný postup pro učení s učitelem je:

- získat seznam vzorků ilustrujících fakt, který je třeba se „naučit“;
- najít vzory v dodaných datech;
- použít nalezené vzory pro klasifikaci nových vstupů;

Mezi klasické algoritmy pro tento přístup patří Bayesovský klasifikátor, Support Vector Machines, rozhodovací stromy nebo neuronové sítě.

Často je také možné setkat se s přístupem, kdy je ručně připravena malá část trénovacích dat a algoritmus již sám dále rozvíjí řešený problém s využitím této předpřipravené části jako podpory.

### 6.1.3 Metody bez učitele

Tento typ metod je založen na předpokladu, že slova s podobným významem se budou vyskytovat ve společnosti podobných slov. Nepoužívá se zde přímé označení kategorií jako například chair/furniture, ale spíše dochází k rozdělení významů do shluků, bez toho, aby byly předem nějak označeny.

Samotné shlukování bývá obvykle prováděno pomocí vektorů a Support Vector Machines s různým způsobem zakódování použitých vektorů.

### 6.1.4 Další metody pro zpřesnění

Během využití výše zmíněných metod je možné ovlivnit úspěšnost rozpoznání pomocí několika metod sledování kontextu:

- Jeden význam pro řeč/kolokaci – Tento postup je založen na předpokladu, že v rámci bloku textu má jedno slovo obvykle stejný význam ve všech výskytech, aby byla zachována konzistence textu.
- Lexikální řetězce – Lexikální řetězec je sekvence souvisejících slov v textu. Je nezávislý na gramatické struktuře a ve výsledku se jedná o seznam slov, zachycujících soudržnost textu. Často se skládá ze slov stejného slovního druhu.

## 6.2 Identifikace větné stavby

Znalost významu jednotlivých slov stále není dostatečná k určení významu celé věty. Například v následujících větách jsou použita stejná slova, ale význam se mění.

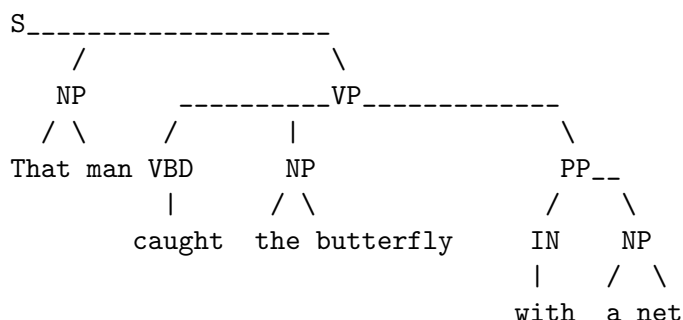
John likes Mary very much.

Mary likes John very much.

Jak je možné pozorovat, význam vět byl ovlivněn pořadím jednotlivých slov. Proto je důležité umět tato významná slova (nebo fráze) ve větě určit a poznat vazby mezi nimi.

Tímto problémem se zabývá syntaxe a rozhodně jej nelze označit za triviální. Obvyklým nástrojem pro popis syntaktických pravidel věty jsou gramatiky (častou používaným typem jsou PCFG pravděpodobnostní bezkontextové gramatiky). V dnešní době se používají dva základní přístupy pro jejich tvorbu:

Prvním ze způsobů je rozklad na větné složky (konstituenty), skupiny slov tvoří fráze, které formují větu. Příkladem může být například věta „That man caught the butterfly with a net“. Celá věta je označena kategorií S, která se pomocí prepisovacích pravidel v gramatice převede na několik frází.



Ve výše uvedeném grafu je použito několik základních typů frází.

- S - sentence - věta
- NP - noun phrase - fráze podstatného jména

- VP - verb phrase - slovesná fráze
- PP - prepositional phrase - předložková fráze

Druhým ze způsobů je závislostní přístup. Struktura věty je zachycena pomocí závislostního stromu, kde uzly odpovídají nejčastěji slovům a hrany zachycují závislosti mezi nimi.

Například ve větě „Sue watched the man at the next table“, jsou fráze Sue a the man závislé na slovesu watched. Obecně je možné říct, že podstatná jména jsou závislá na slovesných frázích a tuto závislost je možné popsat jako sémantickou roli.

Aktuální systémy pro parsing textu obvykle kombinují tyto dva přístupy. Z hlediska konverzačního agenta je potřebné sestavit takovou gramatiku, která nám umožní poznat ve větě několik základních prvků. Těmi jsou podmět, předmět a vazbu mezi nimi. Vhodné by dále bylo i určení času, místa ap.

Bohužel, přirozený jazyk je velmi složitý a sestavit takovouto gramatiku je náročné, přesto existují pokusy toho dosáhnout i za cenu omezení témat pro konverzaci.

Některé existující gramatiky, které se snaží o kompletní pokrytí anglického jazyka jsou například:

- XTAG English Grammar[9]
- The Alvey Natural Language Tools Grammar[4]

Z gramatik s omezenou doménou bych zmínil alespoň ty, které se často vyskytují ve srovnáních:

- ATIS – gramatika pro zpracování dotazů v leteckém provozu
- CommandTalk[5] – tato gramatika byla navržena pro ovládání vojenského simulačního systému
- PT [6] – Gramatika získaná zpracováním Penn Treebank

## 6.3 Určení tématu konverzace

Další informací, která je důležitá pro komunikaci, je aktuální kontext. Jak je možné si všimnout na následujících příkladech, první věta obou konverzacích je shodná, ale má jiný význam, právě díky rozdílnému kontextu.

Flying balloons can be dangerous.  
Sorry, next time I will play outside.

Flying balloons can be dangerous.  
Perhaps, but we have parachute ready at all times.

V prvním příkladu, byla konverzace zaměřena na sportovní náčiní, kdežto ve druhém na dopravní prostředek. Bez znalosti kontextu ovšem není možné rozhodnout, který z významů máme na mysli. Tyto příklady tak ilustrují úzkou vazbu mezi zjištěním kontextu a zjednotnění významu slov.

Existuje několik typů metod pro extrakci kontextu:



- metody založené na lexikálních řetězcích (již zmíněné u zjednodušování významu slov)
- metody využívající externí znalosti pro určení obecného tématu z vazeb mezi slovy v textu
- metody založené na vektorech příznaků pro skupinu dokumentů

### 6.3.1 Metody využívající externí znalosti

Cílem těchto metod je za pomoci sémantické hierarchie identifikovat několik pojmů, které pokrývají téma konverzace. Například z pojmů štěkat, kňučet, srst a zvíře je možné odvodit, že tématem je pes.

Jednou ze sítí, která je často využívána, je sémantická síť WordNet[8] obsahující velké množství sloves, podstatných a přídavných jmen spolu s vazbami mezi nimi.

### 6.3.2 Metody využívající vektory příznaků

Princip těchto metod je založen na využití multidimenzionálních vektorů, reprezentujících jednotlivá témata. Dimenze těchto vektorů reprezentují konkrétní slova, získaná z trénovacích dat a jejich hodnota vyjadřuje míru přítomnosti slova v popisovaném tématu. Slova reprezentující aktuální téma poté získáme nalezením několika vektorů, které se nejvíce podobají právě zpracovávaným datům.

Tato metoda je díky svému jednoduchému konceptu velmi oblíbená a často se používá v systémech pro získávání znalostí (Information retrieval). Také je možné nalezená slova použít jako vodítko pro určení kontextu ve zpracovávaných datech.

## Kapitola 7

# Implementace vlastního agenta

### 7.1 Požadavky

Pro implementaci vlastního komunikačního agenta jsem si stanovil několik cílů. Měl by být schopen identifikovat základní emoce ve vstupu od uživatele, vyhodnotit emoci ve vztahu k sobě a s co nejmenším úsilím (a bez vytváření nějakého většího korpusu) vygenerovat odpověď s použitím některého z výše uvedených algoritmů.

### 7.2 Popis navržené architektury

Celé zpracování vstupu zahrnuje množství kroků, které lze rozdělit zhruba na 6 částí:

- příprava dat
- zpracování větné struktury
- detekce a zpracování emocí
- zapojení emocí do modelu osobnosti
- sestavení odpovědi
- zobrazení odpovědi a vizualizace emocí

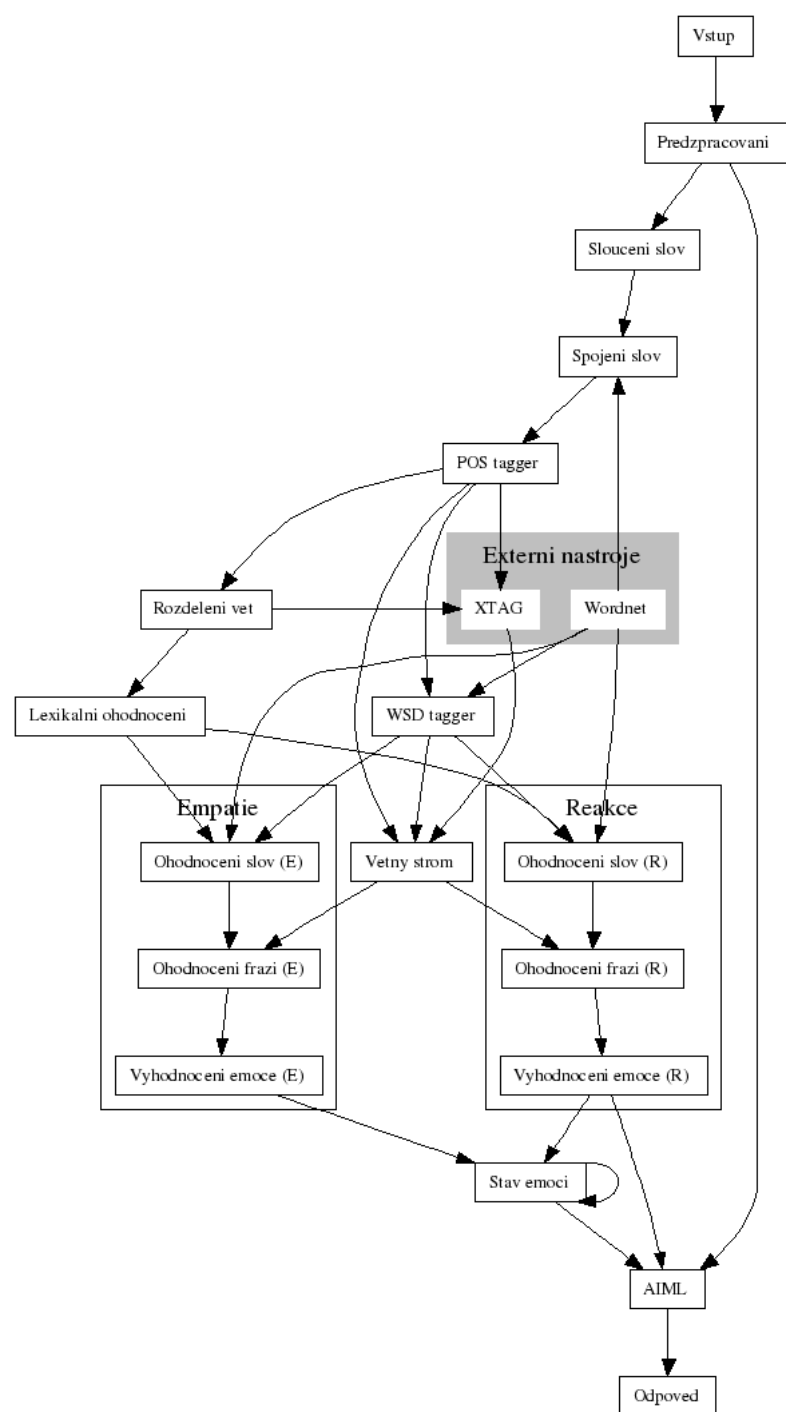
Pro umožnění uživatelských testů je navíc agent navržen jako víceuživatelský a podporuje vzdálený přístup pomocí XML-RPC protokolu.

Schematické znázornění celého postupu je vidět na obrázku 7.1.

Jako programovací jazyk pro implementaci jsem se rozhodl využít Python. Tento jazyk je často používán jak pro rychlé prototypování aplikací, tak pro řešení problémů z oblasti zpracování přirozeného jazyka.

### 7.3 Podpůrné rutiny

Než začnu detailněji popisovat jednotlivé kroky zpracování, je vhodné se zmínit o nástrojích, které se používají na několika různých místech v programu.

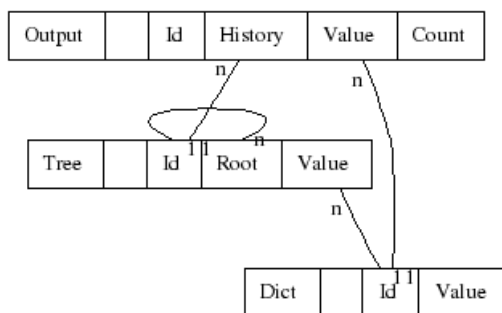


Obrázek 7.1: Schema funkcí agenta

### 7.3.1 DbTree

Jedním z velkých problémů během zpracování přirozeného jazyka je množství dat, které je potřeba uchovat. Proto jsem se inspiroval u struktury TRIE a navrhl vlastní stromovou strukturu, schopnou uchovávat data pro širokou paletu problémů od Ngramového modelu během zjednodoznačování slov až po uložení gramatických pravidel pro použité parsery.

S výhodou jsem využil schopností databázové knihovny Sqlite3, která podporuje podmnožinu jazyka SQL, a navrhl stromovou strukturu ukázanou na následujícím ER diagramu.



Obrázek 7.2: ER diagram

Jak lze vidět, struktura se specializuje na ukládání dat se známou historií, u kterých je potřeba počítat četnost výskytu.

### 7.3.2 Ukládání výsledků

Dalším způsobem jak ulehčit a zrychlit spouštění programu, je ukládání výsledků časově náročných operací, které je potřeba provést pouze jednou. Proto jsem využil standardní knihovny cPickle, která umožňuje serializovat existující struktury na disk a vytvořil třídu LazyLearner, která ve stylu proxy zajišťuje nové vytvoření dat nebo jejich načtení z disku, pokud již existují.

### 7.3.3 Lexikální analýza

Základní kámen většiny nástrojů pro zpracování textu je lexikální analýza. Je to případ i+~+mého programu, i když její využití je omezeno na interpunkci, číselné řetězce a tzv. smile entity.

Protože jsem měl určité nestandardní požadavky (slovo patřící do více tříd zároveň) musel jsem si naimplementovat vlastní analyzátor založený na regulárních výrazech a jejich převodu na konečný automat, který umožňuje koncovým stavům přiřadit výstupní hodnotu.

Je nutné podotknout, že při konstrukci automatu se kvůli jednoduchosti neprovádí minimalizace, ale během vyhodnocení jsou sledovány všechny stavy dosažitelné se vstupním řetězcem. Jediná optimalizace počítaná již při vytváření je epsilon uzávěr, ale aplikován na stavy je až během vyhodnocení.

Konfigurace je načítána z textového souboru. Každý řádek ve formátu „hodnota regularni-vyraz“ je převeden na automat a sloučen s ostatními. Jazyk regulárních výrazů podporuje

všechny běžné operace jako opakování (\*, +), sjednocení (|) a podporuje podvýrazy pomocí kulatých závorek.

### Úryvek konfigurace pro tvorbu automatu

```
-- emoticons
EMO_SADNESS (:|;|„|=)-?(\(|/|\|||[])+
EMO_JOY (B|X|=|:|;|\)(-|^)?(\)|D|]|P|p)+
EMO_DISGUIST 8-?(\||/|\||)+
EMO_SURPRISE (:|=)-?0+
EMO_CONFUSION :-?(S|s)+
EMO_ANGER >(:|=)-?(O|o|E|\(|[])+
```

Analyzátor také podporuje zjištění informace o maximální délce vstupu, při které byl dosažen některý z koncových stavů.

### 7.3.4 Gramatická pravidla

Gramatiky patří k důležitým způsobům zápisu pravidel pro zpracování a analýzu jazyků. Protože přirozené jazyky nejsou výjimkou vytvořil jsem třídu zajišťující práci s bezkontextovými gramatikami a jejich základní úpravy.

Gramatická pravidla jsou načítány ze souboru a umožňují některé specifické operace:

- Základním prvkem gramatiky je pravidlo ve tvaru „neterminál → sekvence terminálů a nonterminálů“ s možností využití znaku |, jak je běžné
- Nonterminály je možné explicitně určit pomocí příkazu „/N Nonterminál“, implicitně jsou za nonterminály označeny všechny tokeny nacházející se na levé straně pravidel
- Pro lepší vlastnosti při zpracování jazyka, jsem navíc definoval nonterminál \*, který funguje jako zástupný pro libovolný jeden token
- Soubory je možné rozdělit do několika menších částí, a ty poté vkládat pomocí příkazu ve formátu „@soubor.txt“
- Protože se na celé řadě míst využívá databáze Wordnet, je umožněna definice nonterminálu přepisujícího se na synonyma všech zadaných slov: „/Wx nonterminal slovo/vyznam|slovo2/vyznam|..“ Kde x je určení slovního druhu (N - podstatné jméno, V - sloveso, A - přídavné jméno).
- Další podobnou funkcí je načtení seznamu pravých stran z externího souboru. Formát příkazu je „/F Nonterminál soubor.txt“ a jeho využití v projektu je například pro seznam vlastních jmen.

Pro další zpracování je potřeba načtené gramatiky převést do nějakého z normovaných tvarů, a proto jsem s přihlédnutím k potřebám CKY parseru zvolil převod na Chomského normální formu, i když vzhledem k charakteru využití gramatik jako klasifikátorů jsem se dopustil jedné nepřesnosti, a to zachování všech variant pravých stran na které se konkrétní levá strana může zredukovat.

### 7.3.5 CKY parser

Samotná gramatika je k ničemu bez parseru, který by ji byl schopný aplikovat na vstup. Protože jsem potřeboval být schopný získat i nedokončené podstromy z parsování (bottom-up) a přitom mít stále dostatečný výkon, rozhodl jsem se implementovat CKY parser.

CKY parser se od jiných algoritmů liší svým inkrementálním způsobem parsování s využitím tabulky, která následně umožňuje zjištění informací o všech nalezených podstromech pro zadanou gramatiku.

Jediným omezením je právě gramatika zadaná pomocí Chomského normální formy, což byl důvod pro její zvolení v předchozí části.

#### Pseudokód algoritmu CKY

- vytvoř tabulku o velikost  $N \times N$  ( $N$  je délka vstupu)
- na řádek 1 doplň do sloupce  $X$  nonterminály  $N$ , pokud existuje pravidlo  $N \rightarrow \text{vstup}[X]$
- pro řádky  $Y$  z  $\langle 2, N \rangle$  prováděj
  - pro sloupce  $X$  z  $\langle 1, 1+N-Y \rangle$  prováděj
    - pro všechna  $D$  z  $\langle 1, Y-1 \rangle$ 
      - do tabulka $[X, Y]$  přidej nonterminál  $N$  pokud existuje pravidlo  $N \rightarrow N_1 N_2$ , kde  $N_1$  je v tabulka $[X, D]$  a  $N_2$  je v tabulka $[X+D, Y-D]$
- pokud je pole tabulka $[1, N]$  neprázdné, vstup odpovídá gramatice

## 7.4 Příprava dat

Před samotným zpracováním vstup je vhodné aplikovat několik transformací, aby další postup byl co nejjednodušší.

Nejprve se celá věta převede na malá písmena, protože další zpracování předpokládá jednotnou velikost písmen (zvláště na Wordnetu závislé metody).

Následně je provedeno nahrazení slov a jejich částí pomocí substituční tabulky. Hlavním úkolem je v této části rozgenerovat zkrácené tvary anglického slovesa být (I'am, you're, we're, ..) a nahrazení některých výrazů používaných ve slangové a „počítačové“ angličtině (dunno, wtf, imho a podobně).

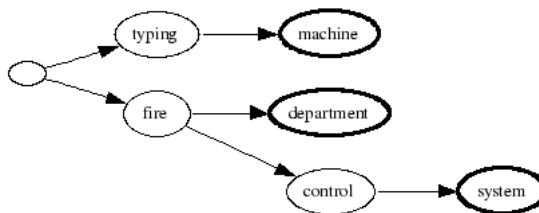
Dále rozdělím vstup na jednotlivé tokeny (slova, interpunkci, emotikony apod.). Pro tuto úlohu využívám lexikální analyzátor popsáný v předchozích částech.

Nakonec provedu spojení a sloučení slov podle definičních tabulek.

Sloučení slov provede přímou konkatenaci řetězců podle lexikální analýzy ve spolupráci s definovanou gramatikou a CKY parserem. Jeho hlavním úkolem je opětovně spojit slova, která byla nesprávně rozdělena předchozími postupy (týká se hlavně nestandardních emotikon).

Pro spojení slov se používá algoritmus využívající DbTree strukturu ve smyslu stromu, kde jednotlivými uzly jsou slova a konkrétní slova nacházející se na cestě mezi koncovým uzlem a kořenem jsou určena ke spojení. Jako oddělovač při spojování se používá znak podtržení. Význam tohoto kroku je v možnosti během dalšího zpracování rozpoznat slovní spojení jako „typing\_machine“ nebo „fire\_department“ a korektně jim přiřadit slovní druh a význam.

Výstupem celého procesu předzpracování je seznam slov a seznam s návodem pro určení jejich slovních druhů.



Obrázek 7.3: Ilustrace spojovacího stromu

## 7.5 Zpracování větné struktury

Ve chvíli kdy máme výsledky z předzpracování, je možné přistoupit k určení slovních druhů, rozpoznání významu jednotlivých slov a nakonec i sestavení derivačního stromu věty s detekcí trojice podmět, přísudek a předmět.

### 7.5.1 Určení slovních druhů

Jednoduchý přehled metod používaných pro určení slovních druhů ve větě jsem uvedl již v první části práce. Pro zapojení do implementace jsem si vybral několik metod v kaskádním zapojení. Tyto metody jsou:

- Konstantní slovní druh (Podstatné jméno)
- Unigramový model
- Bigramový model
- Brillův značkováč

Tyto metody jsem následně učil pomocí Brownova korpusu obsaženého v projektu NLTK.

Určení slovních druhů jednoduše využívá popsané metody. Jedinou výjimkou je existence nápovědy pro rozpoznání, která způsobí obejití těchto metod a přímé převzetí napovězeného slovního druhu.

Například se systém nápověd používá pro mnou navržená značka '*EMO*' znamenající emotikomu.

### 7.5.2 Určení hranic mezi větami

V případě, že vstupní řetězec obsahuje více vět, je třeba tyto najít a oddělit. K tomu používám naivní postup vyhledání sekvence POS značek:

. ( . | : | ; | , | EMO ) \*

Tag '.' v korpusu znamená interpunkci na konci věty. Nalezená sekvence je celá považována za ukončení předcházející věty.

Tabulka 7.1: Konverzní tabulka			
<b>Brown</b>	<b>Penn-Treebank</b>	<b>Brown</b>	<b>Penn-Treebank</b>
BE	VB	BED	VBD
BEDZ	VBD	BEG	VBG
BEN	VBN	BEZ	VBZ
BEM	VBP	BER	VBP
CS	IN	OD	JJ
RBT	RBS	WQL	WRB
QL	RB	QLP	RB
RN	RB	DTI	DT
DTS	DT	ABL	PDT
ABN	PDT	ABX	DT
DTX	DT	AT	DT
AP	JJ	PP\$	PRP\$
PP\$\$	PRP	NP	NNP
NPS	NNPS	NR	NN
NRS	NNS	PN	NN
PPSS	PRP	PPS	PRP
PPO	PRP	PPL	PRP
PPLS	PRP	WPS	WP
WPO	WP		

### 7.5.3 Rozpoznání významu slov

Rozpoznání významu slov je důležitou součástí komunikace. Proto jsem využil své implementace unigramového a trigramového modelu pro významové označování slov ve vstupních datech. Trigramový model výhodně používá výše zmíněný DbTree systém pro uchování dat, protože je lze reprezentovat jako strom o výšce tři, kde listy obsahují pravděpodobnosti výskytu značek. Důvodem pro tuto optimalizaci je značná úspora operační paměti ve srovnání s klasickým řešením zanořenými slovníky.

Jako korpus pro učení těchto metod jsem využil označovaný Brownův korpus z projektu Senseval. Zádrhelem ovšem bylo použití jiných značek pro slovní druhy, než používá část rozpoznávající slovní druhy. Z mně neznámých důvodů byla tato kopie Brownova korpusu ohodnocena značkami typu Penn Treebank. Penn Treebank naštěstí vychází z Brownových značek, a proto bylo možné napsat přibližnou mapovací funkci pro převod. Jako cílovou sadu značek jsem použil právě Penn Treebank, protože existuje zobrazení téměř všech Brownových značek na Penn Treebank, opačně to ale neplatí.

### 7.5.4 Větná stavba

V první části jsem představil důvody syntaktického zpracování věty. Bohužel oba dva parsery pro anglický jazyk, které jsem zkoušel, mají vážné nedostatky, bránící jejich efektivnímu využití.

Parser XTAG neumožňuje jednoduché obejítí svého interního značkovače, a produkuje tudíž mnoho stromů, které neodpovídají slovním druhům zjištěným v předchozích krocích zpra-



cování. Jeho výhodou je, že pro většinu vět je schopen vygenerovat kandidátní stromy prakticky okamžitě.

Dalším parserem, který jsem měl možnost vyzkoušet, byl Charniak. Produkuje lepší výsledky než XTAG, ale bohužel je velmi pomalý.

V aktuálně existující verzi programu je použit XTAG parser, ale zapojení jiného nepředstavuje velký problém, protože je volán jako externí skript s interface ve stylu unixového filtru, tj. používá se standardní vstup a výstup. Formátem rozhraní je žádný, jeden nebo více stromů zapsaných jako S-výrazy.

### Ukázka S-výrazu

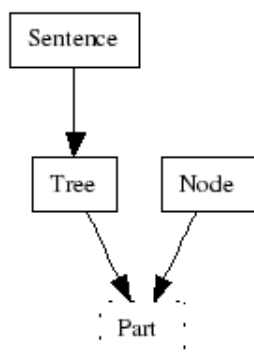
```
(S_r
  (NP (N you))
  (VP_r
    (V are)
    (VP_1 (V_1) (NP (N_r (A selfish) (N_f bastard))))))
```

Tento formát je jednoduchý na zpracování, proto jsem napsal vlastní jednoduchý parser na převod do vnořených seznamů jazyka Python.

Komplikací pro syntaktickou analýzu jsou různá data nepatřící přímo do anglického jazyka, jako například nadměrné použití interpunkce nebo emotikony. Tato data jsou ale potřeba pro další zpracování, a proto jsem musel najít způsob, jak je nejprve odfiltrovat ze vstupu pro parser, ale poté opětovně vložit na správné místo ve výsledném stromu.

Výběr derivačního stromu z kandidátních řešení provádím pomocí dříve získaných slovních druhů, hledám takový strom, kde není žádný konflikt pro podstatná jména, slovesa, přídavná jména a předložky.

Nalezený strom poté rekurzivním sestupem převedu do objektové reprezentace s využitím vlastní hierarchie tříd:



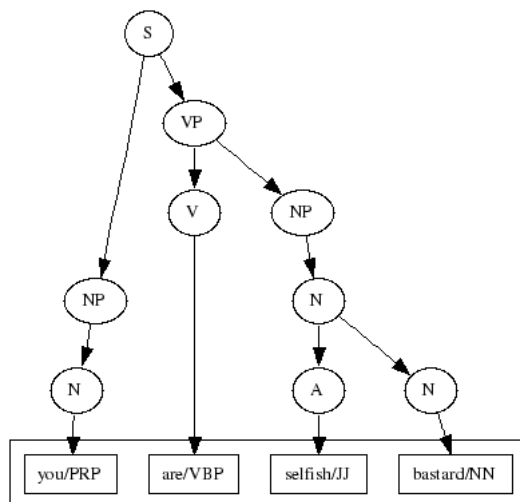
Obrázek 7.4: Hierarchie tříd pro reprezentaci věty

Během tohoto převodu provádím optimalizaci uzlů podle následujících pravidel:

- strom bez listů úplně odstraním;

- strom typu [NP X], kde X není N převedu na X;
- strom typu [VP X], kde X není V převedu na X.

Poté rekurzivním algoritmem provádím sloučení informací o slovních druzích a významech slov a ukládám je k jednotlivým tokenům (třída Node) ve větěném stromu.



Obrázek 7.5: Výsledný strom z S-výrazu

V tomto bodě výpočtu uložím odfiltrované emotikony k tokenu, který se nacházel před nimi, nebo ke kořenu stromu, pokud byly až za značkou pro konec věty.

Dalším krokem zpracování věty je detekce podmětu, přísudku a předmětu pro objekty typu Sentence. Používá se naivní algoritmus předpokládající známou větnou strukturu pro anglický jazyk Subject-Verb-Object, který jsem odvodil z kódu Marka Schmidta. //XXX citace k Marka Schmidta

V běžné anglické větě je obvyklá struktura:

```

(S
  (NP (N podmět))
  (VP (V přísudek)
    (NP předmět))
)

```

Algoritmus proto pracuje s větnými formami a hledá „sourozenecský“ pár NP-VP. Přeskakuje přitom větné formy bez přímého vlivu na strukturu (například předložkové a příslovecné fráze).

## 7.6 Detekce a zpracování emocí

Účelem této práce je prozkoumat možnosti modelování emocí při komunikaci, proto je detekce a zpracování emocí jednou z nejdůležitějších částí práce.

### 7.6.1 Emoční model

Rozhodl jsem se inspirovat popsaným projektem Nemesys a využít zjednodušený model, obsahující pouze šest modelovaných emocí. Jsou to *radost*, *zlost*, *strach*, *znechucení*, *překvapení* a *smutek*. Každá z těchto emocí může nabývat intenzity 0 až 10.

Pro rozhodování ovšem používám více informací, než model Nemesys, protože se pokouším rozpoznat jak emoce projevované konverzačním partnerem (empatie), tak i emoční vliv vstupu na samotného agenta (reakce).

### 7.6.2 Třídy emocí

Během zpracování textu nepracuji přímo s intenzitami emocí, ale využívám symbolická jména pro konkrétní stavy. Každé toto symbolické jméno má potom nadefinovány intenzity emocí a některé související operace, které představuje.

**Ukázka definice**

```
-- CLASS emotion:value emotion:value
MORE more:1
LESS less:1
PRAISE joy:6
INSULT anger:8
DISGUIST disguist:8
FEAR fear:5
PHOBIA fear:10

SAD sadness:8
HAPPY joy:8
ANGRY anger:8
SURPRISED surprise:8

NOT invert:0
CONFUSION joy:4 anger:4 surprise:6
```

Při zpracování množiny těchto symbolických jmen a jejich převodu na výsledné emoce potom postupuji následujícím způsobem:

- pokud je právě zpracovávaná definice emocí, nastav maximum z existující hodnoty a hodnoty v třídě
- pokud je definice příkazem more nebo less, uprav intenzitu (velikost) změny
- pokud je definice příkazem invert, nastav příznak inverze

- pokud byly zpracovány všechny definice, aplikuj změny vynásobením a proved' inverzi, pokud je požadována

Inverze platných emocí (intenzita je větší než 0) se provádí jako doplněk intenzity do hodnoty 10, s výjimkou radosti a zlosti, kde dojde k výměně jejich intenzit.

### 7.6.3 Popis používané gramatiky

V následujících částech budou představeny postupy pro převod vstupního textu na třídy, kterým rozumí výše popsany algoritmus zpracování emocí. Proto je vhodné některé mnou použité třídy představit.

- '*SUBJECT*' nonterminál se rozvíjí na možné identifikace cílové osoby gramatiky (u empatie je to komunikační partner a u reakce samotný program).
- '*POSITIVE*'/'*NEGATIVE*' označuje subjektivní vnímání konkrétního slova jako dobré nebo špatné
- '*WNBELIEVE*' slova a fráze znamenající „věřím, říkám..“, je možné je vypustit ze zpracování, protože celkově mají stejný emoční význam jako to, co za nimi následuje
- '*NOT*' třída invertující hodnotu vypočtené emoce ve větě (způsobem popsáným v kapitole 7.6.2)
- '*MORE*'/'*LESS*' třídy pro změnu intenzity emocí ve větě
- '*JOY*', '*FEAR*', '*SADNESS*' a další třídy aktivující konkrétní emoci

Co se vlastní struktury používané gramatiky týče, je v tuto chvíli tvořena ručně podle průběžně získávaných informací z uživatelského testování. Pro další vývoj bude vhodné vytvořit metodu zpracování označovaného dialogového korpusu, která poskytne stejné informace s co nejmenší námahou.

### 7.6.4 Zpracování na úrovni slov

Předtím, než je možno použít pokročilejší techniky pro emoční zpracování vstupu, je nutné připravit si data na úrovni jednotlivých slov.

Proto se nejprve provede přiřazení tříd pomocí lexikální analýzy popsané výše v části o podpůrných nástrojích.

Tyto třídy jsou následně sloučeny s třídami získanými pomocí aplikace pravidel na zjednoznačená slova.

#### Ukázka pravidel pro zjednoznačená slova

```
-- POS CLASS word/sense|word/sense|...
A~GOOD_WORD nice/2|nice/3|intelligent/4|pretty/1
A~NEGATIVE stupid/3|crazy/1|filthy/3|bad/1|selfish/1
V~HARM kill/1|kill/9|kill/10|harm/1|punch/1|kick/3|beat/2
```

Pravidla jsou definována jako trojice:

- slovní druh (N,V,A);
- název třídy;
- seznam slov ve tvaru slovo/význam oddělených znakem |.

Při jejich zpracování se využívá sémantická síť Wordnet, pro kterou jsou také určeny části definic týkající se konkrétního významu slova. Samotné pravidlo se před uložením rozšíří o všechna slova, která jsou podobná nebo podřazená zadanému významu.

Příkladem může být například definice „A DOG dog/1“, kde dojde k rozšíření na dog, domestic dog, Canis familiaris a všechny názvy ras psů, které jsou ve Wordnet databázi uvedeny.

Samotné uložení poté využívá výše popsaný nástroj DbTree a každá definice je uložena jako:

```
Slovní_druh -> synsetId -> třídy
```

SynsetId je unikátní identifikátor, který Wordnet přiřazuje každému významu, a který je společný všem slovům sdílejícím stejný význam.

Speciálním případem je situace, kdy selže zjednodušení významu pro slovo. Poté se použije naivní přístup a třídy se hledají čistě podle unigramového modelu, který byl vytvořen zároveň s učením výše popsaných pravidel. Tento model přiřadí slovu všechny třídy, pozorované u stejně vypadajících zjednodušených slov. Je i uložen ve stejném souboru, jen interně používá formát:

```
__NAIVE -> Slovo -> třídy
```

### 7.6.5 Zpracování na úrovni celé věty

Ve chvíli kdy již máme ohodnocena jednotlivá slova, je možné začít hledat významné fráze. Pro jejich definici jsem využil modulu pro bezkontextové gramatiky popsaného mezi nástroji.

Kromě nonterminálů popsaných explicitně v pravidlech, jsou v gramatice k dispozici ještě všechny Part-Of-Speech značky a jejich skupiny (NOUN, VERB, ADVERB a podobně).

Třídy detekované zadanou gramatikou odpovídají jednotlivým nonterminálům v jejím popisu.

Zpracování frází poté, v případě že je k dispozici, využívá ještě informací ze syntaktického parseru věty, zapouzdřených v objektovém modelu Sentence.

Jako parser pro nalezení podstromů je využit, již dříve popsaný, CKY parser. V jeho prvním kroku jsou, k předchozím algoritmem získaným nonterminálům, přidány ještě odpovídající třídy zjištěné v předchozích částech zpracování (lexikální analýza a detekce na úrovni slov).

Další úpravou vstupních dat je přidání prázdného tokenu s třídou \_START na začátek a s třídou \_END na konec.

Z výsledných produktů CKY parseru je potom zajímavá vyplněná tabulka, kde každá neprázdná hodnota určuje jeden nebo více podstromů, které odpovídají gramatice.

Prvotní mapování tříd do objektového modelu, je provedeno pomocí rekurzivního průchodu stromem typu „preorder“. Pro každý podstrom i list ve stromu je uchovávána pozice počátečního a koncového tokenu. Na základě této informace se zjistí maximální délka podstromu (počet tokenů) i jeho počátek ve vstupu. Toho lze následně využít jako indexů do získané CKY tabulky, pro identifikaci třídy celého podstromu.

### 7.6.6 Sémantické zpracování

Asi nejnáročnější úlohou při zpracování je vyrovnat se s možnými chybami v předchozích krocích a pokusit se stanovit emoční význam celé věty.

Nejjednodušší metodou, kterou jsem zkoušel bylo získání všech emočně zabarvených tříd, které se ve větě vyskytují a hromadné sloučení jejich emocionálních ohodnocení. Protože tento způsob zřejmě nebude přesný, využil jsem derivačního stromu, získaného v předchozí kapitole pro test další možnosti.

V metodě s využitím syntaktické analýzy proto funguje rozdělení na věty podle spojek (and, or) a detekce negace. Spojky způsobí sjednocení emocí ze všech vět. Negace v podstromu způsobuje inverzi zjištěné emoce. Dále se aplikuje sloučení emocí pouze v rámci podstromů, čímž by mělo být možné odstranit některé chyby předchozí popsané metody.

V další práci předpokládám možnost určení emocí přímo pomocí trojice podmět, přísudek a předmět, a jim odpovídajících podstromů.

## 7.7 Model osobnosti

Veškeré atributy týkající se správy emocí, schopnosti odpovídat, báze znalostí a podobně jsem sdružil do třídy Personality, která je zodpovědná za modelování osobnosti.

Osobnost je složena z několika základních částí:

- informace o sobě – sem patří například jméno, informace o tvůrci, definice oblíbených a neoblíbených předmětů a ostatní pevné součásti, používané v modulu pro generování odpovědi
- znalosti o okolí získané během konverzace
- nastavení míry empatie, reakce, stálosti nálady a útlumu emocí
- aktuální stav krátkodobých a dlouhodobých emocí

Tato část je také zodpovědná za změny emocí podle informací získaných v předchozích krocích.

Pro zpracování zjištěných emocí jsem navrhl dva algoritmy.

Aktuálně používaný algoritmus sloučí emoce na základě lineární funkce typu:

```

změna = aktuální emoční stav * útlum emocí
        + míra empatie * empatie
        + míra reakce * reakce

```

```

nálada = (stálost nálady * nálada + změna) / (stálost nálady + 1)
aktuální emoční stav = (aktuální emoční stav * stálost + změna) / (stálost + 1)

```

Složitější algoritmus využívá neuronových sítí pro mapování vstupních dat na výstupní. Tato metoda má pravděpodobně větší potenciál, ale je náročná na kvalitu a množství, trénovacích dat, proto jsem ji ponechal pro možné rozšíření.

## 7.8 Sestavení odpovědi

Pro sestavení odpovědi podle vstupu a zpracovaných emocí jsem využil modifikovaný interpret jazyka AIML. AIML je dialekt XML, obsahující definice vzorů otázek, odpovědí na ně a několik dalších příkazů pro jednoduché uchování kontextu a znalostí o okolí.

Jeho zjednodušené schéma je možné popsat takto (kompletní popis lze nalézt v dokumentaci na oficiálních WWW stránkách):

```

AIML := <aiml> CATEGORIES* </aiml>
CATEGORIES := <category> PATTERN TEMPLATE </category>
PATTERN := <pattern> TEXT </pattern>
TEMPLATE := TEXT | RANDOM
RANDOM := <random> ITEM* </random>
ITEM := <li> TEXT </li>

```

Moje rozšíření spočívala v přidání atributů pro omezení emocí, ve kterých je možné zadanou odpověď použít.

Následující atributy se týkají značek *'template'* a *'li'*:

- *empathy* – předpokládaný emoční stav komunikačního partnera podle posledního vstupu;
- *reaction* – reakce agenta na aktuální vstup.

Tag *'li'* umožňuje poté i výběr pomocí aktuálního emočního stavu agenta.

Všechny atributy jsou textové a obsahují výčet názvů emocí oddělených mezerou pro který je daná značka platná.

### Ukázka emocionálního AIML

```

<aiml>
  <category>
    <pattern>How are you?</pattern>
    <template><random>
      <li state='joy'>I'm OK. Thank you.</li>
      <li state='joy'>I feel fine.</li>
    </template>
  </category>
</aiml>

```

```

        <li state='anger disgust'>Not very well...</li>
        <li state='fear'>Scared beyond imagination.</li>
        <li state='surprised'>I am really surprised.</li>
    </random></template>
</category>
</aiml>

```

Zdrojové kódy obsahují experimentální nástroj pro ohodnocení 'li' značek atributem 'state' podle pravidel pro zpracování empatie.

### 7.8.1 Algoritmus hledání odpovědi

Před samotným vyhledáním odpovědi na uživatelem zadaný vstup jsou provedeny operace na zjištění dominantní emoce v reakci a empatii. Navíc je také pomocí neuronové sítě získána informace o emocích, které by se v odpovědi neměly projevit (kvůli přirozenosti projevu).

Dominantní emocií je taková emoce, která má aktuálně nejvyšší hodnotu a není nulová.

Samotné vyhledání odpovědi probíhá jako vyhledávání ve stromu, se vstupním řetězcem zkonstruovaným jako

```
[vstupní text, THAT, that text, TOPIC, topic text,
PERCEPTION, reakce, EMPATHY, empatie]
```

Do dalšího zpracování se předají data obsažená za značkou 'TEMPLATE', která se nachází co nejbližší konci vstupního řetězce. Těmito daty je obvykle struktura obsahující přímo text nebo značky z AIML, které se následně vyhodnotí.

Vyhodnocení značky 'random' jsem upravil s ohledem na výběr odpovědi s preferovanými emocemi. Proto se u všech značek 'li' se zadanými emočními hodnotami vyhodnocuje zda neodporují aktuálním emočním požadavkům. Vznikne-li situace, že není možno vybrat žádný z výstupů, aplikuje se původní sémantika a dojde k vybraní náhodné odpovědi ze seznamu.

## 7.9 Zobrazení odpovědi a vizualizace emocí

Aplikace má grafické uživatelské rozhraní vytvořené pomocí knihovny Tkinter v jazyce Python. Okno se skládá ze dvou hlavních částí: historie proběhlé konverzace a zobrazení emocí.

Z těchto částí je zajímavá právě část zabývající se zobrazením emočního stavu. Jednou z jejích částí je totiž dynamicky generované schematické znázornění obličejových prvků, které imitují mimiku lidského obličeje.

Použité prvky a některé polohy, kterých mohou nabývat, jsou znázorněny na obrázcích níže:

Řídící logika pro toto zobrazení je tvořena jednovrstvou bipolární perceptronovou sítí o 60 vstupech a 39 výstupech. Vstup je kódován jako šest skupin po deseti vstupech, každá





skupina vyjadřuje intenzitu jedné z emocí. Hodnota vstupů ve skupině je potom nastavena tak, že pro intenzitu  $N$  je nastaveno prvních  $N$  vstupů na 1 a zbytek na -1.

Výstupy mají stejný způsob zakódování hodnot ve skupině, s tím, že je potřeba při dekódování interpolovat hodnoty, kde není souvislá řada aktivních výstupů. Výstupy jsou uspořádány do následujících skupin (v závorce počet výstupů ve skupině):

- poloha levého obočí (5+5)
- poloha pravého obočí (5+5)
- výška levého oka (5)
- výška pravého oka (5)
- velikost očí (5+5)
- poloha úst (5)
- tvar úst (pozitivní, negativní)
- otevření úst
- šířka úst (malá, velká)
- výška úst (malá, velká)

## 7.10 Rozšíření pro účely testování

Jako rozšíření vhodná pro ladění a testování jsem implementoval možnost vzdáleného přístupu ze samostatného klienta a možnost uložení grafické reprezentace derivačního stromu zpracované věty.

### 7.10.1 Vzdálený přístup

Pro síťovou komunikaci využívám bezstavový protokol XML-RPC s tím, že exportuji jen minimum potřebných funkcí. Tyto funkce jsou `getSession`, `closeSession` pro správu spojení a funkce `callback` pro odeslání vstupního textu a získání odpovědi spolu s emočním stavem.

Samotný protokol komunikace je potom již velmi jednoduchý:

### Postup komunikace

```
Client          xml-rpc event          Server

-----> getSession ----->
<----- session ID -----

enter sentence
-----> callback -----> |
<----- response, emotions -----> | repeat for every sentence
vizualize response          |
-----> score -----> |

-----> closeSession ----->
<----- OK ----->
```

#### 7.10.2 Vizualizace větné stavby

Nejjednodušším způsobem, jak získat grafickou reprezentaci používaných struktur, bylo zapojit do zpracování knihovnu Graphviz. Vstupem je jednoduchý jazyk pro popis grafu a výstupem může být jeden z mnoha podporovaných formátů, já jsem zvolil PNG.

#### Ukázka zápisu grafu z obrázku 7.3

```
digraph TRIE {
    rankdir=LR;
    node [width=.3,height=.2,fontsize=10];

    machine [style=bold];
    department [style=bold];
    system [style=bold];
    S;

    S -> typing;
    typing -> machine;
    S -> fire;
    fire -> department;
    fire -> control;
    control -> system;
}
```

Pro Python existuje knihovna PyDot, která tento jazyk objektově zapouzdřuje, takže s využitím rekurzivního sestupu na objektovém modelu věty jsem schopen vygenerovat jeho grafickou reprezentaci (obrázek 7.5).

## Kapitola 8

# Testy a vyhodnocení

Je velice obtížné vyhodnotit tak nezměřitelnou oblast, jakou jsou emoce. Proto se v uživatelských testech zaměřím alespoň na informace o míře shody mezi vnímáním samotných uživatelů, metodami využívajícími syntaktickou analýzu a postupy, kde se pokročilé techniky nepoužívají.

### 8.1 Uživatelské testy

Pro tento účel jsem upravil uživatelské rozhraní programu, aby umožňoval s odesláním věty přímo zadat i emoční hodnotu věty podle uživatele (jeho náladu). Poté jsem zaznamenával rozpoznané empatie a reakce před a po použití techniky syntaktické analýzy věty.

Další sledovanou veličinou bylo subjektivní hodnocení uživatele pro shodu odpovědi a projevené emoce (bodový systém 0–10).

V následujících tabulkách jsou výsledky experimentů. Čísla udávají počet zpracovaných vět s vlastnostmi určenými řádkem a sloupcem tabulky. Počty správných rozpoznání jsou v tabulkách zvýrazněna.

Celková úspěšnost pro metody bez využití pokročilých technik NLP (tj. bez syntaktické analýzy) byla 51% a při jejím využití 37%. Pokud zanedbáme případy, kde nebyla na vstupu žádná emoce, tak získáme bez využití syntaktické analýzy úspěšnost 43%, ale při jejím

Tabulka 8.1: Výsledky porovnání uživatelského vstupu (řádky) a rozpoznané empatie (sloupce)

	none	joy	anger	fear	sadness	disguist	surprise
none	<b>86</b>	19	2	2	1	3	0
joy	14	<b>71</b>	5	0	0	0	1
anger	34	3	<b>25</b>	6	2	4	1
fear	5	0	0	<b>8</b>	1	0	0
sadness	18	3	8	0	<b>9</b>	2	0
disgust	34	4	10	0	2	<b>18</b>	0
surprise	17	7	0	0	0	0	<b>7</b>

Tabulka 8.2: Výsledky porovnání uživatelského vstupu (řádky) a rozpoznané empatie (sloupce) s využitím syntaktické analýzy

	none	joy	anger	fear	sadness	disguist	surprise
none	<b>101</b>	6	1	1	1	2	1
joy	62	<b>29</b>	0	0	0	0	0
anger	54	2	<b>14</b>	0	1	3	1
fear	9	0	0	<b>5</b>	0	0	0
sadness	33	3	1	0	<b>2</b>	1	0
disgust	56	3	1	0	0	<b>8</b>	0
surprise	25	3	0	0	0	0	<b>3</b>

Tabulka 8.3: Výsledky porovnání uživatelského vstupu (řádky) a rozpoznané empatie (sloupce)

	none	positive	neutral	negative
none	<b>86</b>	19	1	7
positive	14	<b>71</b>	1	5
neutral	35	10	<b>16</b>	10
negative	73	7	6	<b>71</b>

využití pouze 19%. Tento propad je pravděpodobně způsoben nedostatečným pokrytím jazyka gramatickými pravidly.

Určitého zlepšení je možné dosáhnout při omezení množství rozpoznávané informace na pozitivní, negativní a neutrální emoce. Jako negativní se považují všechny mnou používané emoce, kromě *překvapení* a *radosti*. *Radost* je považována za pozitivní a *překvapení* za neutrální. Úspěšnosti rozpoznání pro jednodušší metodu se pohybují cca o 5% výše.

Poslední tabulka znázorňuje uživatelské hodnocení odpovědí a agentem projevených emocí z pohledu na realističnost.

Bohužel je velice náročné získat potřebné množství kvalitních dat pro jednoznačné vyhodnocení. Je možné si všimnout, že uživatelé málokdy využívali vět s emocemi jako *strach* nebo *smutek*. Také *překvapení* zřejmě bylo často pouze v kombinaci s *radostí* (v tabulkách je vidět velký překryv).

I z těchto výsledků je ovšem poznat, že agent má ještě velké mezery v definičních pravidlech a gramatikách pro hodnocení věty. Další informací, kterou je možno z dat odvodit je, že

Tabulka 8.4: Výsledky porovnání uživatelského vstupu (řádky) a rozpoznané empatie (sloupce) s využitím syntaktické analýzy

	none	positive	neutral	negative
none	<b>101</b>	6	2	4
positive	62	<b>29</b>	0	0
neutral	58	6	<b>5</b>	2
negative	119	5	2	<b>31</b>

Tabulka 8.5: Hodnocení realističnosti odpovědí

body	-	1	2	3	4	5	6	7	8	9	10
vět	85	3	6	2	4	7	2	7	16	8	51

pokročilejší techniky na zpracování jazyka jsou citlivější na trénovací data a pro ty, které jsem měl k dispozici byly úspěšnější jednodušší metody.

## Kapitola 9

### Závěr

V této práci jsem ukázal aktuální stav na poli konverzačních agentů z hlediska zpracování jazyka a modelování osobnosti, kde i přes několik desítek let trvající výzkum těchto oblastí stále ještě není možné ani napodobit komunikaci člověka s člověkem, natožpak naučit stroje porozumět a vést smysluplnou konverzaci.

Dále jsem se pokusil nastínit některé problémy a postupy, které považuji za důležité pro rozvinutí konverzačních schopností dnešních chatterbotů. Jedná se hlavně o témata související se zpracováním přirozeného jazyka jako zjednoznačnění významu slov, určení sémantických vazeb ve větách a zpracování kontextu.

V dalších částech jsem předvedl implementaci vlastního agenta, založenou na existujících technikách a v uživatelských testech poukázal na její slabiny a potíže se získáním trénovacích dat.

Z cílů, které jsem si stanovil na začátku práce, jsem byl schopen splnit všechny. Podstatná ovšem byla míra úspěšnosti, která se bohužel nepohybovala příliš vysoko, a pro jejich zlepšení bude zřejmě třeba dalšího výzkumu.

Dalšími možnými cíli a tématy ke zpracování jsou proto zejména postupy pro automatizované získávání trénovacích dat a postupy jak z nich strojově sestavit gramatiky pro rozpoznávání emocí. Pro vylepšení komunikační části agenta by bylo vhodné se zaměřit na metody prohledávání vzorů, definovaných v AIML, a sofistikovanější propojení kontextu mezi AIML a prováděným zpracováním jazyka.

Tato práce byla velkou zkušeností se zpracováním jazyka a emocí a pomohla mi pochopit a vyzkoušet si některé důležité techniky.

Také jsem získal zkušenosti s prezentací práce v konferenčním prostředí, protože moje práce byla přijata do sborníku soutěžní studentské konference EEICT 2007[15].

Zbývá tedy ještě mnoho práce, které bude potřeba vykonat, aby bylo možné naučit stroje komunikovat přirozenějším způsobem.

# Literatura

- [1] Five-factor model. Dostupný z <http://www.personalityresearch.org/bigfive.html> (platné 12/2006).
- [2] A. Collins A. Ortony, G. Clore. *The Cognitive Structure Of Emotions*. Cambridge University Press, 1998.
- [3] C. Bartneck. Integrating the occ model of emotions in embodied characters. In *Proceedings of the Workshop on Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges*, 2002. Dostupný z [http://www.bartneck.de/work/bartneck\\_hf2002.pdf](http://www.bartneck.de/work/bartneck_hf2002.pdf) (platné 11/2006).
- [4] T. Briscoe C. Grover, J. Carrol. The alvey natural language tools gramman, 1993. Dostupný z <http://www.cl.cam.ac.uk/research/nl/anlt.html>.
- [5] Artificial Intelligence Center. Ct grammar, 1999.
- [6] E. Charniak. Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1031–1036. MIT Press, 1996.
- [7] Manfred Eckschlager, Regina Bernhaupt, and Manfred Tscheligi. Nemesys: neural emotion eliciting system. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1347–1350, New York, NY, USA, 2005. ACM Press.
- [8] Ch. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. Další informace také na <http://wordnet.princeton.edu/>.
- [9] X. Group. A lexicalized tree adjoining grammar for english, 1995.
- [10] A. Kubík. *Intelligentní Agenty*. Computer Press, 2004.
- [11] M. Kukačka. Modely emocí pro autonomní agenty a umělé bytosti, 2005. Dostupný z <http://ksvi.mff.cuni.cz/brom/eseje/Emoce2005.pdf>.
- [12] R. Kurzweil. *The Age of Intelligent Machines*. The MIT Press, 1990.
- [13] Michael L. Mauldin. Chatterbots, tinymuds, and the turing test: entering the loebner prize competition. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 16–21, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.
- [14] Bruce J. Neubauer. Designing artificial personalities using jungian theory. *J. Comput. Small Coll.*, 20(1):297–305, 2004.

- [15] M. Sivák. Emotional modelling in communication agents. In *Proceedings of the 13th Conference STUDENT EEICT 2007*, volume 2, pages 266–268. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií a Fakulta informačních technologií, 2007.
- [16] Alan Turing. Computing machinery and intelligence. *Mind*, LIX(236), 1950.
- [17] Juan D. Velasquez and Pattie Maes. Cathexis: a computational model of emotions. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 518–519, New York, NY, USA, 1997. ACM Press.